Soffid LDAP

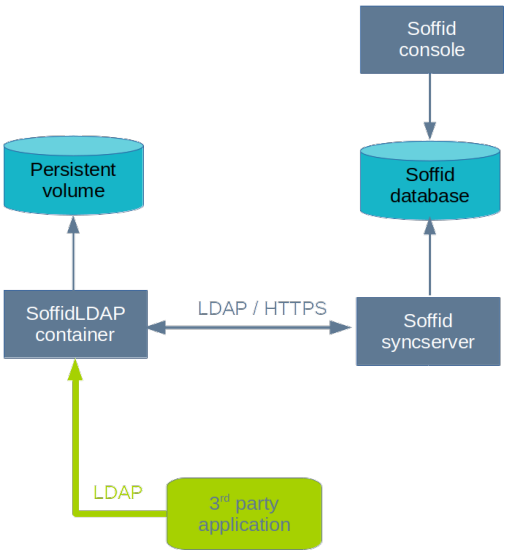
Soffid LDAP

- Introduction to Soffid LDAP
- How to install Soffid LDAP?
- How to debug Soffid LDAP?
- Connecting to the LDAP using user short names
- Seamless authentication
- Soffid overlay configuration

Introduction to Soffid LDAP

What is Soffid LDAP?

Soffid LDAP is a complete LDAP Server based on OpenLDAP. It is always distributed as a docker container. The proposed system architecture is as follows:



The SoffidLDAP is not directly using Soffid database. Instead, it is using its own database, that can contain a subset or superset of the information stored in Soffid database.

The integrations work as follows:

- Soffid syncserver pushes to Soffid LDAP any change in Soffid database. It does this task by means of the standard LDAP connector. Thus, you can perform any kind of transformation or filtering during this process.
- Soffid syncserver can also fetch any change performed in SoffidLDAP by means of the standard LDAP connector
- Soffid LDAP delegates any password management task to Soffid sync server. This includes validation against trusted third-party authenticators and checking new password policies.
- Third-party applications can access Soffid LDAP by using LDAP (or LDAPS) protocol.

How to install Soffid LDAP?

Installation with docker

Prerequisites

To install **Soffid LDAP**, you must install Docker. Despite Docker desktop can be used for testing purposes, Docker container runtime usage is recommended.

Installation

To start Soffid LDAP, execute:

```
docker volume create ldapconf
docker volume create ldapdata
docker run --name soffidldap -p 1389:389 -v ldapconf:/etc/ldap/slapd.d -v ldapdata:/var/lib/ldap -d -e
SOFFID_SERVER=https://<SYNCSERVERNAME>:760 -e SOFFID_AGENT=<SOFFID_AGENT> -e
USER=<ADMIN_USERNAME> -e PASSWORD=<ADMIN_PASSWORD> -e DN=<YOUR_BASE_DN>
soffid/soffidldap:latest
```

Image

```
gbuades@gbuades-ThinkPad-T590:~$ docker run --name soffidldap -p 1389:389 -v ldapconf:/etc/ldap/slapd.d -v ldapdata:/var/lib/ldap -e S
FFID_SERVER=https://localhost:760 -e SOFFID_AGENT=ldap -e USER=cn=admin -e PASSWORD=SuperSecret -e DN=dc=soffid,dc=com soffid/soffidld
p
Performing initial setup
Creating director : dc=soffid,dc=com
Administrator user: cn=admin,dc=soffid,dc=com
Initial password  : SuperSecret
Soffid server     : https://localhost:760
Soffid agent      : ldap
Creating configuration
Register MDB engine
Register Soffid Overlay
Starting Service
Creating database
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "olcDatabase=mdb,cn=config"

modifying entry "olcDatabase={-1}frontend,cn=config"

Install Soffid overlay
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
*** soffid: Initialised with syncserver: https://soffid.bubu.lab:760
*** soffid: Initialised with syncserver: https://soffid.bubu.lab:760
*** soffid: Initialised with syncserver: https://soffid.bubu.lab:760
adding new entry "olcOverlay=soffid,olcDatabase={1}mdb,cn=config"

Install MemberOf overlay
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=module,cn=config"

adding new entry "olcOverlay=memberof,olcDatabase={1}mdb,cn=config"

adding new entry "olcOverlay=refint,olcDatabase={1}mdb,cn=config"

Creating root DC object
adding new entry "dc=soffid,dc=com"

DONE
```

Now, you can connect to the LDAP Server using the user name and password used to create the docker instance.

Docker commands

Stop the service

```
docker stop soffidldap
```

Start the service

```
docker start soffidldap
```

Remove the service

```
docker rm soffidldap
```

Docker Volumes

- The initial configuration files are located in the **ldapconf volume**, usually in /var/lib/docker/volumes/ldapconf/_data
- The initial data files are located in the **ldapdata volume**, usually in /var/lib/docker/volumes/ldapdata/_data

Installation with docker compose

Prerequisites

To install **Soffid LDAP** using **Docker Compose**, you must first have installed it.

Installation

Create a new **folder** or use the one in your current environment; you need to add the following service to the **docker-compose.yaml** file.

> Remember to enter the correct values for variables of type **<VALUE>**.

```
services:

  soffidldap:
    image: soffid/soffidldap:latest
    environment:
      SOFFID_SERVER: https://<SYNCSERVERNAME>:<PORT>
      SOFFID_AGENT: <SOFFID_AGENT>
      USER: <ADMIN_USERNAME>
      PASSWORD: <ADMIN_PASSWORD>
      DN: <YOUR_BASE_DN>
    networks:
      - network
    ports:
      - 1389:389
    volumes:
      - ldapconf:/etc/ldap/slapd.d
      - ldapdata:/var/lib/ldap

networks:
```

```
  network:
    name: netversion4
    driver: bridge

volumes:

  ldapconf:
    name: soffidldap_ldapconf
  ldapdata:
    name: soffidldap_ldapdata
```

This is an example of a configuration.

```
soffidldap:
  image: soffid/soffidldap:latest
  environment:
    SOFFID_SERVER: https://sync-server-version4:1768
    SOFFID_AGENT: SoffidLDAP_dockerCompose
    USER: cn=admin
    PASSWORD: Soffid4.
    DN: dc=soffid,dc=local
  networks:
    - network
  ports:
    - 1390:389
  volumes:
    - ldapconf:/etc/ldap/slapd.d
    - ldapdata:/var/lib/ldap
```

We deploy the service.

```
docker compose up -d
```

Testing

Now, you can connect to the **LDAP** Server using the **user** and **password** used to create the docker service.

> You can use a free tool such as **Apache Directory Studio**.

Docker compose commands

>

> If you need further information about Docker Compose, you can visit the Docker Compose commands page

Docker Volumes

- The initial configuration files are located in the **ldapconf volume**, usually in /var/lib/docker/volumes/ldapconf/_data
- The initial data files are located in the **ldapdata volume**, usually in /var/lib/docker/volumes/ldapdata/_data

How to debug Soffid LDAP?

Installation

Prerequisites

To install **Soffid LDAP**, you must install Docker. Despite Docker desktop can be used for testing purposes, Docker container runtime usage is recommended.

Steps to debug

1.- First of all, you have to add the **DEBUG environment variable** to your container

```
-e DEBUG: 65535
```

2.- You need to restart the Soffid LDAP container

```
sudo docker restart <YOUR_CONTAINER>
```

3.- In this point, you can check your LDAP

4.- Retrieve the log

5.- Delete the DEBUG environment variable

6.- Restart the Soffid LDAP container

Log level

Multiple log levels may be used and the levels are additive. To display what numbers correspond to what kind of debugging, consult the table below. The possible values for <integer> are:

Level	Description
-1	enable all debugging
0	no debugging

1	trace function calls
2	debug packet handling
4	heavy trace debugging
8	connection management
16	print out packets sent and received
32	search filter processing
64	configuration processing
128	access control list processing
256	stats log connections/operations/results
512	stats log entries sent
1024	print communication with shell backends
2048	print entry parsing debugging
16384	syncrepl consumer processing
32768	only messages that get logged whatever log level is set

The desired log level can be input as a single integer that combines the (ORed) desired levels.

Connecting to the LDAP using user short names

Connecting to the LDAP

The Soffid LDAP accepts the usual way to connect to the LDAP service using the distinguished name and password for the user.

Additionally, one can use the uid attribute to login. Then for the user:

```
dn: cn=user,ou=test,dc=soffid,dc=com
cn: user
givenName: John
sn: Snow
uid: jsnow
userPassword: secret
```

The user can use one of these two ways to login to the LDAP server:

1. USER: cn=user,ou=test,dc=soffid,dc=com PASSWORD: secret
2. USER: uid=jsnow,dc=soffid,dc=com PASSWORD: secret

Seamless authentication

The password authentication process is redesigned to allow users to connect to Soffid LDAP using their internally stored password, or a password trusted by Soffid.

1. The first attempt is to check the password against the **local userPassword attribute**. If it fails, Soffid LDAP will connect any of the configured sync servers, and will let the sync server validate the password.
2. Then, the sync server will check the password against **Soffid internal tables**.
3. If the password is not accepted according to Soffid internal tables, the authentication request will be forwarded to any **trusted target systems**. The trusted flag is enabled or disabled on a per-agent basis, at the agents configuration page
4. If everything fails, the login is rejected

Anyway, whenever a password change is detected by Soffid, the attribute userPassword can be updated. This is done by means of the LDAP connector.

The opposite way also works for password changes. Whenever the user attribute userPassword is updated, SoffidLDAP does:

1. Generates a secure hash using SSHA-256, and replaces the provided value for the corresponding hash
2. Notifies Soffid sync server the password has been changed
3. Adds an additional value {SOFFID} to the userPassword attribute. Now, the userPassword attribute has two values: the SSHA hash and the Soffid marker
4. Soffid sync server, in turn, validates the received passwords and stores it in Soffid database and other target systems. This only happens if the trusted flag is enabled for the Soffid LDAP agent

Soffid overlay configuration

The communication channel from Soffid LDAP to Soffid sync server is configured by the Soffid overlay. The overlay configuration object looks like this:

```
#
# Soffid overlay
dn: olcOverlay=soffid,olcDatabase={1}mdb,cn=config
olcOverlay: soffid
objectClass: olcOverlayConfig
objectClass: olcSoffidConfig
olcDomain: soffidldap
olcSoffidServer: https://soffid.bubu.lab:760
```

The two configuration parameters are olcDomain and olcSoffidServer.

olcSoffidServer is a multivalued attribute that hosts the list of Soffid sync servers that will receive password change notifications. They will be used for authentication purposes as well.

olcDomain is the agent name used in Soffid console to manage the current Soffid LDAP Database